一种分布式语义增强的词汇链文本表示模型构建方法

曲云鹏 [1,2,3]　王文玲 [3]
[1](中国科学院大学　北京 100049)
[2](中国科学院文献情报中心　北京 100190)
[3](国家图书馆　北京 100081)

摘要: 【目的】利用分布式语义关联计算词衔接关系，解决目前词汇链构建时存在的词间关系探测深度不够等问题，提高词汇链构建质量。【方法】对词汇链构建的技术方法进行归纳，利用 WordNet 词典关系来计算文本中语言单元的语义关联，利用分布式记忆模型来计算语言单元之间的潜在语义关系，将这两种语义关系结合起来实现词汇链文本表示模型的构建。同时在理论研究的基础之上选择医学领域科技论文进行对比实验。【结果】在文本主题描述方面，本文方法的词汇链构建结果要优于非贪婪算法，算法耗时与非贪婪算法相当。【局限】算法耗时较长；没有完整考虑词衔接关系；只在对医学领域科技文献的主题识别中验证了该方法的有效性，还需要在更多领域进行证明。【结论】分布式语义关联可以识别潜在语义，对使用多元短语构建词汇链也有较大的帮助，能有效地增强词汇链构建效果。

关键词: WordNet　分布式记忆　词汇链　分布式语义
分类号: TP393　G350

1　引　言

词汇链(Lexical Chain)文本表示模型是一种对语篇中的词汇衔接(Lexical Cohesion)关系进行建模的文本表示模型，能够体现语篇中丰富的语义信息。词汇链构造了一个易于理解的上下文环境，有助于确定多义词在文本中的具体含义；词汇链能为文本结构以及文本一致性提供线索，有助于理解文本的大意。词汇链文本表示模型结构简单，广泛应用于文本切分[1]、自动摘要[2]、文本过滤[3]、自动问答[4]、拼写错误识别[5]和情感识别[6]等领域。

词汇衔接关系的计算方法可以归为三类：基于词典的方法、基于统计的方法和基于图的方法[7]。基于词典的词汇链构建方法使用词典中定义好的语义关联关系来计算词汇衔接关系，易于理解、便于实施，在词汇链构建过程中得到了最广泛的应用，是构建词汇链的主要方法。针对英文文献，主要使用 WordNet、罗杰词典(Roget's Thesaurus)等进行构建[8-9]。针对中文文献，主要使用 HowNet、《同义词词林》等进行构建[10-12]。基于统计信息的词汇链构建方法对围绕主题时词汇同时出现的这种倾向性进行统计语言学分析形成同现关系知识库，然后利用知识库计算对象文本的相似度来表示词汇衔接关系，从而构建词汇链。所使用的算法主要包括基于极的重叠聚类算法[13]、LDA 方法[14]、E 指数方法[15]等。基于图的方法将文本转化为图，然后利用图聚类等方法寻找词汇链[16]。由于基于词典和基于统计信息的方法二者互补，因此开始有人尝试将两类方法结合起来构建词汇链，如 Marathe 等尝试将分布式语义和词典相结合，在词汇链构建中进行应用[17]，获得了不错的效果。

通讯作者：曲云鹏，ORCID：0000-0002-1611-0904，E-mail：quyp@nlc.cn。

对词汇链构建方法进行研究和归纳后，发现目前词汇链构建方法中词衔接计算方法中存在一些问题。

(1) 使用词典可以探测到明确的语义关联，使用统计信息可以探测到词之间的潜在关联，二者都是词衔接中的重要类型。但是目前使用的统计信息相对有限，无法更深入地探测候选词之间的潜在关联。

(2) 候选词的上下文信息对候选词词义或词间关系计算的影响较大，但是目前对候选词上下文的使用仍然有限。

(3) 尽管已经有研究尝试将词典和统计信息融合使用，但是仍没有解决词典中未收录的词或者短语无法参与词汇链构建的问题。

基于以上分析，笔者提出一种分布式语义增强的词汇链构建算法，尝试解决以上提到的问题。在算法中，利用 WordNet 词典关系来计算文本中语言单元的语义关联，利用分布式记忆模型来计算语言单元之间的潜在语义关系，并对二者进行融合计算，构建词汇链。本文所提方法的特点为:

(1) 保留原文本中的更丰富的信息

本研究提出了分布式语义加强的词汇链构建方法，同时计算候选词之间的语义关系和分布式语义关系，从多个角度对候选词之间的关联进行计算，可以发现更丰富的语义，保留原语篇中更多的信息。

(2) 考虑了上下文环境对于术语含义的影响

本文方法中将计算候选词的分布式语义关联强度纳入到词汇链构建的过程中。在计算过程中，尽可能充分考虑候选词所在上下文的环境，包括候选词在训练集所处环境中的介词搭配情况、连词搭配情况以及形容词和动词的使用情况。这些信息对于候选词的消歧和词衔接关系的识别有很重要的参考作用。

(3) 词典中未收录的词或短语也可以参与构建词汇链

本方法中可以通过计算这些候选词或者候选短语的分布式语义关联和共现关联来计算关系，因此在词汇链结果中，也将出现很多短语或者专业词汇。

2 分布式记忆模型

分布式语义模型(Distributional Semantics Models, DSM)的基本理论是语言学领域的分布式假设理论，即“在相同的上下文中出现的词汇在某种程度上有类似的含义”[18]。在这种假设下，一个词可以映射为分布式语义空间中的一个向量，向量的维度对应词周围的上下文环境，维度值可以通过与上下文环境共现信息来确定。如果两个词所对应的向量较为相似，那么这两个词就有相似的含义[19]。分布式语义模型的建立过程为收集术语在语料库中的上下文环境并进行分析，通过计算术语和文档、上下文中语言单元或者句法结构的共现信息，将术语所在的语言环境表示为一个多维的向量空间，建立术语–文档矩阵、术语–上下文矩阵、词对–模式矩阵等，从而建立起分布式语义空间[20]。通过这种空间模型可以体现术语之间的语义关联，可以计算语言单元之间的相似度，进一步发现语言单元之间潜在的语义联系。较为知名的分布式语义模型包括潜在语义分析(Latent Semantic Analysis)[21]、随机索引(Random Indexing)[22]、依赖向量(Dependency Vector)和分布式记忆(Distributional Memory)[23]等。

在几种分布式语义模型中，分布式记忆空间在规则的设定、三元组的使用方面都比较灵活，因此本文选择分布式记忆模型来计算候选词之间的分布式语义相似度。通过设定抽取规则，分布式记忆模型可以从术语上下文中抽取共现信息，表示为“术语–关联–术语”三元组，同时计算每个三元组的权值，构成一个三维的张量<术语，关联，术语，值>。与其他分布式语义框架不同的是，分布式记忆模型中的关系可以进行自由设定，可以选择句法关系(如介词关系)，也可以选择其他任何一种可连接两个术语的关联类型。另外，分布式记忆模型可以根据需要将三维张量转化为不同类型的二维分布式矩阵，如“<术语 1, (关系，术语 2)>”矩阵或者“<(术语 1，术语 2，)，关系>”矩阵，从而从不同的角度来表现文本[24]。目前，分布式记忆模型已经得到了广泛应用，英语、德语[25]和克罗地亚语[26]的分布式记忆库已经开始构建，并应用于多种自然语言的处理。

3 分布式语义增强词汇链构建算法

分布式语义增强词汇链构建算法的主要步骤有:构建候选词列表、语义关联关系计算、分布式语义关系计算、关系融合计算、词汇链构建，如图 1 所示。其中需要解决的关键问题包括分布式语义空间的构建、分布式语义关系的计算、语义关联的计算、关系

融合的计算和词汇链的构建算法。

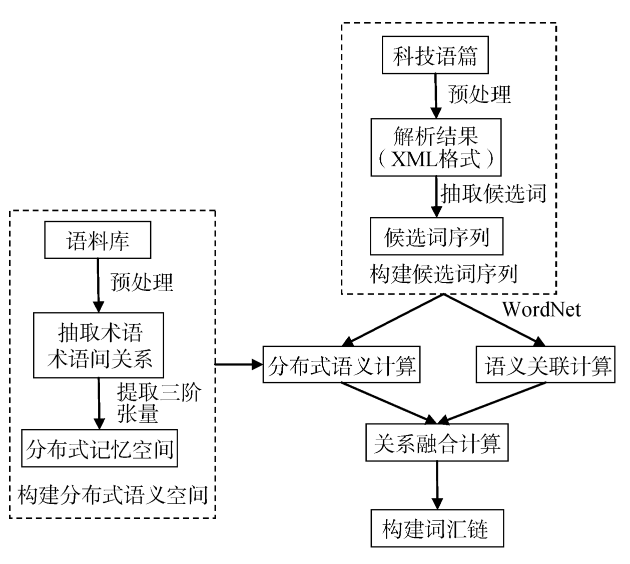


图 1 分布式语义增强词汇链构建流程图

3.1 分布式语义空间的构建和分布式语义关系的计算

分布式语义空间的构建首先需要从语料库中识别术语和术语之间的关系，组合成为三元组后，再计算本地互信息(Local Mutual Information, LMI)值，构成分布式语义空间。

对语料进行词性识别和依赖语法解析，选择类型为"NN，NNS，NNP，NNPS"的名词和依赖语法中类型为"Compound"的二元短语[27]，作为术语；在依赖语法解析结果中，选择介词、连词、形容词和动词 4 种关联规则，作为三元组中的关联，抽取三元组<术语，依赖语法类型，术语>，规则具体内容如表 1 所示:

表 1 三元组抽取规则

关联规则名称	依赖语法类型	举例
介词规则	pobj	<damage, pobj, Cardiomyocute>
连词规则	conj	<heart failure, conj, event>
形容词规则	acomp、amod、nmod	<death, amod, sudden>
动词规则	agent、dobj、iobj、subj	<cure, dobj, disease>

抽取三元组之后，使用本地互信息(LMI)公式[28]计算三元组的关系权重，如果 LMI 值为负值，则舍弃该组合。LMI 计算公式如下:

$$LMI = P(x,r,y)\log\frac{P(x,r,y)}{P(x)P(r)P(y)} \tag{1}$$

这样三元组就转化为一个带权重的三阶张量<术语，依赖语法类型，术语，LMI>，当所有的三元组都转化为带权重的三阶张量之后，分布式记忆空间则构建完成。

3.2 词衔接关系的计算

首先对目标文档进行预处理，采取与分布式记忆空间相同的术语抽取方法，选择类型为"NN，NNS，NNP，NNPS"名词和依赖语法中类型为"Compound"的二元短语，作为构建词汇链的候选词。

本文提出的方法中，需要计算两个候选词之间的分布式语义关联和词典语义关联。

(1) 分布式语义关联的计算

计算候选词的分布式语义关联时，需要动态地从分布式语义空间中抽取候选词的环境向量。分布式语义空间中，术语的上下文环境以三阶张量<术语 1，依赖语法类型，术语 2, LMI>的方式保存。在进行抽取时，用(依赖语法类型，术语 2)作为候选词 x 的环境向量维度，将三阶张量转为二阶矩阵<x, (r, y)>，矩阵中的值为对应的 LMI 值。表 2 中为术语"death"和"heart failure"通过转化后的二阶向量。

表 2 二阶向量示意表

术语 \ 维度	nmod, inclusion	dobj, report	$amod^{-1}$, sudden	dobj, worsen	conj, event	……
death	25.6134	84.9131	427.8113	0	0	……
heart failure	0	0	0	44.3085	82.158	……

最后通过计算两个向量间的夹角余弦值来表示两个候选词的潜在语义关系强度，计算结果直接参与词汇链的计算，余弦值计算公式[29]如下所示:

$$\cos\theta = \frac{\sum_{i=1}^{n}(x_i \times y_i)}{\sqrt{\sum_{i=1}^{n}(x_i)^2} \times \sqrt{\sum_{i=1}^{n}(y_i)^2}} \tag{2}$$

(2) 语义关联关系的计算

本算法中语义关联关系的计算选用英文通用词典 WordNet 作为工具[30]。计算时参考 Silber 等的方法[31]，选择重复、同义词/反义词、整体/部分关系、上下位类、兄弟关系这 5 种语义关系，窗口距离设定为 1 个句子、3 个句子和不限距离，每种情况下对关系赋予不同权值，用于词间关系的计算。权重赋值取决于词汇的语义关联类型及其窗口距离，具体赋值如表 3 所示。

表 3　WordNet 中语义关系的权重赋值

语义关系 \ 窗口距离	1 个句子	3 个句子	不限距离
重复	1	1	1
同义词/反义词	1	1	1
整体/部分关系	1	0.5	0.5
上下位类	1	0.5	0.5
兄弟关系	1	0.3	0

3.3　关系融合计算

两个候选词之间的词汇衔接关系强度需要将两种关系强度进行融合计算。经过实验发现，计算中采用加权融合方法较为合理，公式如下：

$$\mathrm{Relation}(w_i, w_j) = a \times \mathrm{Wordnet}(w_i, w_j) + b \times \mathrm{Dist}(w_i, w_j) \quad (3)$$

其中，$\mathrm{Wordnet}(w_i, w_j)$为语义关联关系强度，$\mathrm{Dist}(w_i, w_j)$为分布式语义关系强度。a 和 b 为经验常数。

3.4　词汇链的构建算法

候选词与已有词汇链之间的词汇衔接关系强度取候选词和链中所有词的词汇衔接关系强度的平均值，公式如下：

$$\mathrm{Relation}(w_i, \mathrm{Chain}) = \mathrm{average}(\mathrm{Relation}(w_i, w_n)) \quad (4)$$

构建词汇链时，参考 Barziley 等的方法[32]，依照候选词出现的顺序对候选词逐一进行处理，先计算该候选词与现有所有词汇链的词汇衔接关系强度 $\mathrm{Relation}(w_i, \mathrm{Chain}_j)$，若当前词汇链为空或者所有 $\mathrm{Relation}(w_i, \mathrm{Chain}_j)$都小于设定阈值 Thres(w, C)，则新建以当前候选词开头的词汇链，否则将该候选词加入 $\mathrm{Relation}(w_i, \mathrm{Chain}_j)$值最大的词汇链 Chain_j。构建词汇链的计算机伪码如下所示：

```
候选词汇链序列 LC_List()
For 每个候选词
  For 每个词汇链
    计算候选词与链的关系权值 Relation(w, Chain)
  End for
  If (LC_List() is empty) or (all score (w, C)<Thres(w, C))
    建立以当前候选词开头的词汇链
  Else
    加入关系权值最大的词汇链
  End if
End for
```

4　实验和分析

采用医学领域的科技论文作为实验数据，以“heart”和“cardiac”为关键词在 ScienceDirect 数据库中进行检索，从检索结果中选择 100 篇英文全文文档作为分布式记忆的语料库，从而进行分布式语义空间的构建。构建完的分布式语义空间中共有 71 023 个三元组。使用斯坦福大学研制的自然语言分析处理工具包 Stanford CoreNLP[33]对语料进行预处理，包括词性标注、停用词处理等，将文本转化为程序可自动处理的 XML 标准格式文档。

实验中，设定词间关系计算公式中的两个经验参数 a 和 b 取值为 1，如公式(5)所示。候选词加入词汇链的阈值 Thres(w, C)设置为 0.5。

$$\mathrm{Relation}(w_i, w_j) = \mathrm{Wordnet}(w_i, w_j) + \mathrm{Dist}(w_i, w_j) \quad (5)$$

4.1　通过关键词识别效果进行质量对比

从关键词抽取结果角度对词汇链构建结果的质量进行对比。本文以“heart”和“cardiac”为关键词在 ScienceDirect 数据库中进行检索，从检索结果中随机选择 50 篇摘要信息。请一位医学专家对 50 篇摘要进行阅读，每篇标注 3-6 个关键词。随后，请另一位医学专家分别对非贪婪算法构建的词汇链和本文算法构建的词汇链结果进行审阅，根据词汇链的构建情况完成关键词抽取。将抽取结果对照专家给出的关键词，计算其准确率和召回率。结果如表 4 所示：

表 4　算法主题识别对比表

算法	准确率	召回率
本文算法	70.43%	73.82%
非贪婪算法	52.92%	57.51%

从表 4 结果中可以看出，本文算法的词汇链构建结果的准确率和召回率要高于非贪婪算法构建结果。

4.2　分布式语义对于词汇链构建的影响分析

从结果中随机抽取 5 个样本进行数据统计，结果如表 5 所示。通过对数据结果进行分析，可以发现，本文算法在语义信息发现数量、词语含义的确定和候选词发现数量三个方面优于非贪婪算法。

(1) 分布式语义发现了更为丰富的语义信息。

本文算法结果中，发现有效 WordNet 关系 3 225 个，计算分布式语义关联共 14 710 次，得到结果大于 0 的分布式语义关联 9 508 个，无 WordNet 关联但是有分布式语义关联的术语对共 6 803 个，其中二元短语参与计算的有效分布式语义关联 347 个。在无 WordNet

表 5　词汇链构建结果主要数据对照表

算法	样本名	样本长度	发现的候选词数量	二元短语数量	有效词汇链数量	有效词汇链的长度	有效词汇链中包含的候选词数量	有效结果中包含的二元短语数量
分布式语义增强词汇链构建算法	样本 1	291	117	13	8	34, 12, 10, 8, 7, 5, 5, 5	86	13
	样本 2	364	143	21	11	18, 15, 15, 14, 14, 11, 8, 7, 6, 5, 5	118	21
	样本 3	313	117	12	10	13, 10, 10, 9, 8, 6, 6, 5, 5, 5	77	10
	样本 4	347	127	17	7	43, 23, 15, 8, 7, 6, 5	107	17
	样本 5	283	128	19	4	36, 26, 7, 7	76	19
非贪婪算法	样本 1	291	98	—	6	15, 8, 8, 6, 5, 5	47	—
	样本 2	364	117	—	5	11, 10, 9, 6, 5	41	—
	样本 3	313	100	—	4	20, 8, 8, 5	41	—
	样本 4	347	109	—	7	16, 14, 13, 10, 7, 6, 5	71	—
	样本 5	283	92	—	4	22, 6, 6, 6	40	—

关联但是存在分布式语义关联的术语对中，分布式语义关联较强的术语对包括<baseline function, impairment>、<patient, treatment>、<correlation, difference>、<artery, disease>等，这些术语对的分布式语义关联强度都在 0.5 左右。对这些术语对的原语篇进行人工阅读分析，发现这些术语对在语篇中的确存在较强的关联，但是使用 WordNet 无法发现这些关联。可以说，分布式语义所发现多元短语和潜在语义关联对于词汇链构建有很明显的影响，很大程度上弥补了只借助词典进行词汇链构建的缺陷。

(2) 分布式语义可以根据上下文环境分析候选词的含义，进而更准确地发现词衔接关系。

在词汇链构建过程中，分布式语义在确定候选词含义方面起到了作用，可以更准确地发现词衔接关系。如“evolution”一词，含有两个含义，第一指进化，第二指进展。在使用非贪婪算法进行词汇链构建时，算法选择了第二个含义，将“evolution”同“action”划分在同一个词汇链。而使用本文算法构建词汇链时，通过分布式语义计算发现“evolution”同“origin”的关联更强烈。类似的例子还有“species”的含义应为“物种”，相比“model”，“species”的含义同“human”更接近，进而分为一个词汇链等。

(3) 分布式语义帮助本文方法发现更多的候选词。

在候选词的总数方面，本文算法在 5 篇测试样例中总共发现候选词 632 个、二元短语 82 个，在最终的有效词汇链中保留了 464 个候选词、80 个二元短语；非贪婪算法总共发现候选词 516 个，在最终的有效词汇链中保留了 240 个候选词，候选词数量明显少于本文算法。

4.3　算法耗时

实验中，对 5 个样本进行词汇链构建时耗费的时间如表 6 所示。分布式语义增强算法和非贪婪算法的时间复杂度一致，但是分布式语义增强算法需要从分布式语义空间中实时抽取环境向量进行相似度计算，大大增加了构建的时间。

表 6　算法耗时对比(ms)

算法	样本 1	样本 2	样本 3	样本 4	样本 5
非贪婪算法	37 531	81 425	145 708	74 170	36 828
分布式语义增强算法	6 416 513	11 615 083	11 582 981	16 000 995	7 145 408
建立索引后的分布式语义增强算法	77 723	167 048	136 472	88 515	61 987

通过研究发现，在分布式语义空间稳定的情况下，两个术语在该空间中的分布式语义关联是稳定的。因此，在实验中将经常使用的术语之间的分布式语义关联保存在数据库中，作为分布式语义关联计算的索引。该方法提高了实际使用中的算法效率，算法计算时间达到了和非贪婪算法相当的水平。

5　结　语

本研究的创新工作主要体现在以下两点:

(1) 提出一种分布式语义增强的词汇链构建方法

在该方法中，采用分布式语义关联对候选词之间的语义关系进行加强，在词汇链构建时可以考虑更多更丰富的文本关联，探测到隐藏于深层次中的词衔接

关系，提高了词汇链的构建效果。通过实验可以看出，本文提出的分布式语义增强的词汇链构建方法在实验结果中优于非贪婪算法，计算过程中所发现的分布式语义关系对词汇链的构建也产生了足够的影响，提高了词汇链构建的效果。

(2) 提出一种分布式记忆模型的应用场景

分布式记忆模型是一个新颖的模型，在国内目前还缺少有效的研究。本研究率先在词汇链的构建中使用分布式记忆模型，根据需要设定了三元组抽取规则，提出了一种分布式记忆模型的使用场景，并在实验中验证了其效果。为今后的分布式记忆模型研究打下了基础。

在未来的工作中，还需要解决以下的一些问题。分布式语义解决的是在大规模语料中发现候选词的潜在关联的问题，但是无法解决在某一个文献中的候选词有特定的含义和特殊的语义关联，因此对目标文献进行词汇共现分析获取其中的语义关联，可作为对分布式语义增强方法的进一步补充。词衔接关系的发现仍然不足。Hoey 针对词汇链的理论基础——词衔接关系，进行了研究，提出词衔接关系的6种类型[34]，而目前还只能探测到其中的 3 种类型，需要其他的方法来更全面地探测词衔接关系。在下一步的工作中，可以尝试使用文献计量学中表示关键词之间关联强度的统计指数 Salton 指数和 Jaccard 指数来计算词汇共现关系，作为对分布式语义增强方法的补充。同时对分布式语义进行更深入的研究，使得词汇衔接关系的计算更为完整。

参考文献：

[1] Manabu O, Takeo H. Word Sense Disambiguation and Text Segmentation Based on Lexical Cohesion [C]. In: Proceedings of the 15th Conference on Computational Linguistics-Volume 2. Stroudsburg: Association for Computational Linguistics, 1994: 755-761.

[2] Barzilay R, Elhadad M. Using Lexical Chains for Text Summarization [A]. // Mani I, Maybury M T. Advances in Automatic Text Summarization[M].Cambridge: MIT Press, 1999: 357-380.

[3] Li S, You W, Li T, et al. Lexical-chain and It's Application in Text Filtering [C]. In: Proceedings of the International Conference on Information Technology: Coding and Computing. Washington: IEEE Computer Society, 2004: 288-292.

[4] Moldovan D, Novischi A. Lexical Chains for Question Answering [C]. In: Proceedings of the 19th International Conference on Computational Linguistics-Volume 1. Stroudsburg: Association for Computational Linguistics, 2002: 1-7.

[5] St-Onge D. Detecting and Correcting Malapropisms with Lexical Chains [D]. Toronto: University of Toronto, 1995.

[6] Naveen Kumar M, Suresh R. Emotion Detection Using Lexical Chains [J]. International Journal of Computer Applications, 2012, 57(4): 1-4.

[7] 曲云鹏，王文玲. 词汇链文本表示模型计算方法综述[J]. 知识管理论坛，2016(2): 136-144. (Qu Yunpeng, Wang Wenling. An Overview on the Computing Method of the Lexical Chain Text Representation [J]. Knowledge Management Forum, 2016(2): 136-144.)

[8] Hirst G, St-Onge D. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms [J]. Lecture Notes in Physics, 1995, 728(9): 123-149.

[9] Morris J, Hirst G. Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text[J]. Computational Linguistics, 1991, 17(1): 21-48.

[10] 刘铭，王晓龙，刘远超. 基于词汇链的关键短语抽取方法的研究[J]. 计算机学报, 2010, 33(7): 1246-1255. (Liu Ming, Wang Xiaolong, Liu Yuanchao. Research of Key-Phrase Extraction Based on Lexical Chain [J]. Chinese Journal of Computers, 2010, 33(7): 1246-1255.)

[11] 胡学钢，李星华，谢飞，等. 基于词汇链的中文新闻网页关键词抽取方法[J]. 模式识别与人工智能，2010，23(1): 45-51. (Hu Xuegang, Li Xinghua, Xie Fei, et al. Keyword Extraction Based on Lexical Chains for Chinese News Web Pages[J]. Pattern Recognition and Artificial Intelligence, 2010, 23(1): 45-51.)

[12] 裘江南，罗志成，王延章. 基于词汇链的应急预案主题抽取方法研究[J]. 情报学报，2008，27(6): 891-896. (Qiu Jiangnan, Luo Zhicheng, Wang Yanzhang. Research on Semantic Relatedness Based Subjects Extraction from Emergency Plans Literature [J]. Journal of the China Society for Scientific and Technical Information, 2008, 27(6): 891-896.)

[13] Dias G, Santos C, Cleuziou G. Automatic Knowledge Representation Using a Graph-based Algorithm for Language-independent Lexical Chaining [C]. In: Proceedings of the Workshop on Information Extraction Beyond the Document. Stroudsburg: Association for Computational Linguistics, 2006: 36-47.

[14] Remus S, Biemann C. Three Knowledge-free Methods for Automatic Lexical Chain Extraction [C]. In: Proceedings of NAACL-HLT 2013. Stroudsburg: Association for Computational Linguistics, 2013: 989-999.
[15] 叶春蕾, 冷伏海. 基于词汇链的路线图关键词抽取方法研究[J]. 现代图书情报技术, 2013(1): 50-56. (Ye Chunlei, Leng Fuhai. Study on the Keyword Extraction from Roadmap Based on the Lexical Chains [J]. New Technology of Library and Information Service, 2013(1): 50-56.)
[16] Medelyan O. Computing Lexical Chains with Graph Clustering [C]. In: Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop. Stroudsburg: Association for Computational Linguistics, 2007: 85-90.
[17] Marathe M, Hirst G. Lexical Chains Using Distributional Measures of Concept Distance [C]. In: Proceedings of the 11th International Conference on Computational Linguistics. 2010: 291-302.
[18] Basili R, Pennacchiotti M. Distributional Lexical Semantics: Toward Uniform Representation Paradigms for Advanced Acquisition and Processing Tasks [J]. Natural Language Engineering, 2010, 16(4): 347-358.
[19] Molino P, Basile P, Caputo A, et al. Exploiting Distributional Semantic Models in Question Answering [C]. In: Proceedings of the 2012 IEEE 6th International Conference on Semantic Computing. Washington, DC: IEEE Computer Society, 2012: 146-153.
[20] Padó S, Lapata M. Dependency-based Construction of Semantic Space Models [J]. Computational Linguistics, 2007, 33(2): 161-199.
[21] Landauer T K, Dumais S T. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge [J]. Psychological Review, 1997, 104(2): 211-240.
[22] Sahlgren M. An Introduction to Random Indexing [C]. In: Proceedings of Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, Copenhagen, Denmark. 2005.
[23] Baroni M, Lenci A. One Distributional Memory, Many Semantic Spaces [C]. In: Proceedings of the Workshop on Geometrical Models of Natural Language Semantics. Stroudsburg, PA: Association for Computational Linguistics, 2009: 1-8.
[24] Baroni M, Lenci A. Distributional Memory: A General Framework for Corpus-based Semantics [J]. Computational Linguistics, 2010, 36(4): 673-721.
[25] Padó S, Utt J. A Distributional Memory for German [C]. In: Proceedings of the KONVENS 2012. 2012: 462-470.
[26] Šnajder J, Padó S, Agić Ž. Building and Evaluating a Distributional Memory for Croatian [C]. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics.2013: 784-789.
[27] De Marneffe M-C, Manning C D. Stanford Typed Dependencies Manual [EB/OL]. [2016-04-07]. http://nlp.stanford.edu/software/dependencies_manual.pdf.
[28] Evert S. The Statistics of Word Cooccurrences [Elektronische Ressource]: Word Pairs and Collocations [D]. Stuttgart: University of Stuttgart, 2005.
[29] Turney P D, Pantel P. From Frequency to Meaning: Vector Space Models of Semantics [J]. Journal of Artificial Intelligence Research, 2010, 37(4): 141-188.
[30] Fellbaum C, Miller G. WordNet: An Electronic Lexical Database [M]. Cambridge, MA: MIT Press, 1998.
[31] Silber H G, Mccoy K F. Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization [J]. Computational Linguistics, 2002, 28(4): 487-496.
[32] Barzilay R, Elhadad M. Using Lexical Chains for Text Summarization [C]. In: Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization. 1997: 10-17.
[33] Manning C D, Surdeanu M, Bauer J, et al. The Stanford CoreNLP Natural Language Processing Toolkit [C]. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2014: 55-60.
[34] Hoey M. Patterns of Lexis in Text [M]. Oxford University Press, 1991.

作者贡献声明:

曲云鹏: 提出研究思路, 设计研究方案, 进行实验, 起草论文;
王文玲: 采集、清洗和分析数据, 论文修订。

利益冲突声明:

所有作者声明不存在利益冲突关系。

支撑数据:

支撑数据[1,3]见期刊网络版 http://www.infotech.ac.cn; 支撑数据[2]由作者自存储, E-mail: quyp@nlc.cn。

[1] 曲云鹏. 支撑数据.xlsx. 本文 4.1 节的关键词质量对比数据.
[2] 曲云鹏. 代码.zip. 本文中涉及的源代码.

[3] 曲云鹏. 语料和结果.zip. 算法测试用的语料及生成的词汇链结果.

收稿日期: 2016-04-08
收修改稿日期: 2016-06-23

Using Semantic Model to Build Lexical Chains

Qu Yunpeng[1, 2, 3]　Wang Wenling[3]
[1](University of Chinese Academy of Sciences, Beijing 100049, China)
[2](National Science Library, Chinese Academy of Sciences, Beijing 100190, China)
[3](National Library of China, Beijing 100081, China)

Abstract: [**Objective**] This paper uses Distributional Semantics to build high quality lexical chains. [**Methods**] First, we built an algorithm using WordNet Thesaurus to compute the semantic relations among language units of the texts. Second, we adopted the Distributional Memory Model to compute their latent semantic relations. Finally, we combined these relations to build the lexical chains, which were examined with papers from medical science. [**Results**] The proposed algorithm was better than the non-greedy methods to describe the papers' topics. [**Limitations**] The efficiency of the algorithm needs to be improved. It should also be examined with papers from other fields. [**Conclusions**] The proposed model can detect the latent semantic relation, and then improve the quality of lexical chains building with phrases.

Keywords: WordNet　Distributional Memory　Lexical Chain　Distributional Semantics